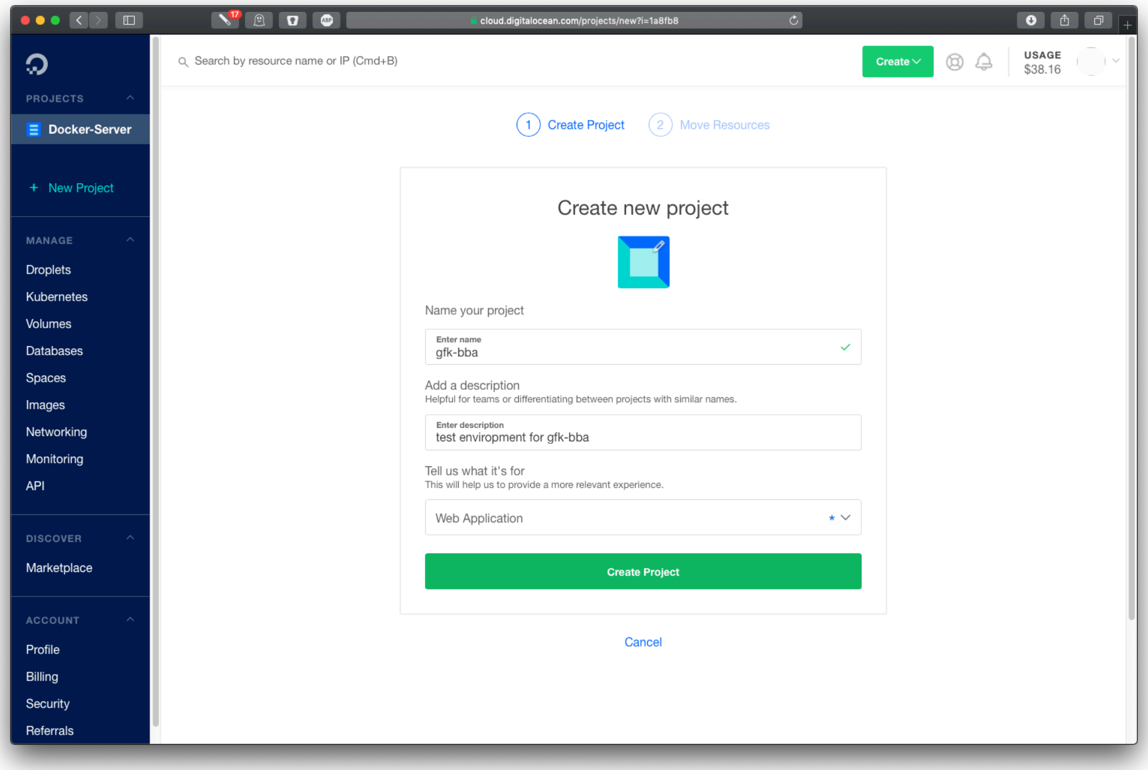
Setup GfK-BBA using Docker on a blank server

For this example we use Digital-Ocean, but any hosting provider should be fine.

In Digital Ocean it’s best practice to create a project first and then put resources in this project (a small virtual server “droplet” is all we need)



Create new project

Name your project

Enter name
gfk-bba

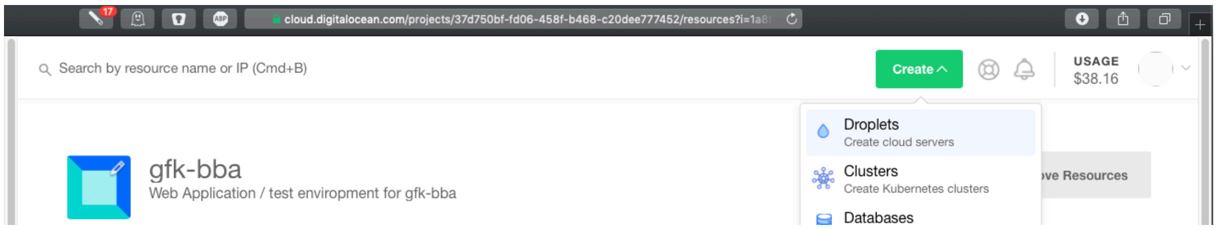
Add a description
Helpful for teams or differentiating between projects with similar names.

Enter description
test enviropment for gfk-bba

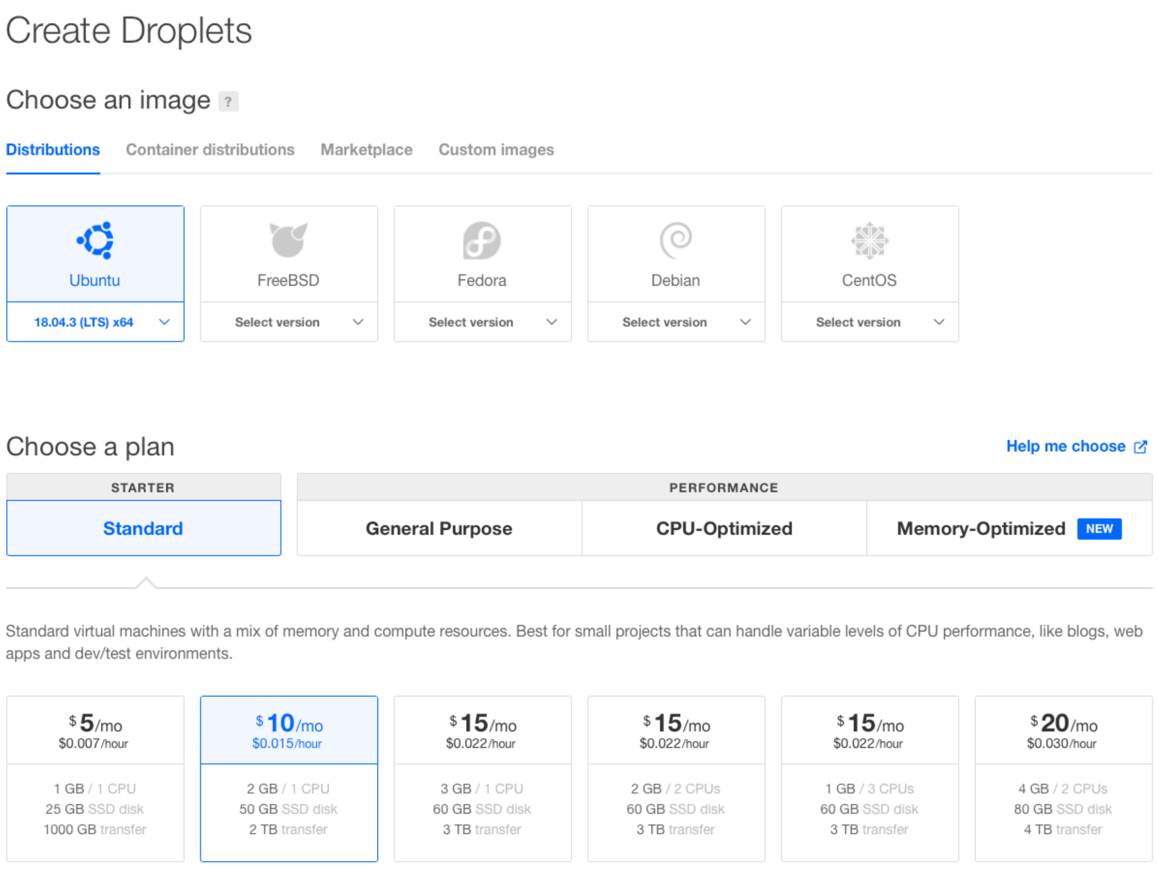
Tell us what it's for
This will help us to provide a more relevant experience.

Web Application

Next step: we create a droplet (a virtual server)


cloud.digitalocean.com/projects/37d750bf-fd06-458f-b468-c20dee777452/resources?i=1a8
Search by resource name or IP (Cmd+B)
Create
USAGE
$38.16
Droplets
Create cloud servers
Clusters
Create Kubernetes clusters
Databases
gfk-bba
Web Application / test envirpnment for gfk-bba
ve Resources

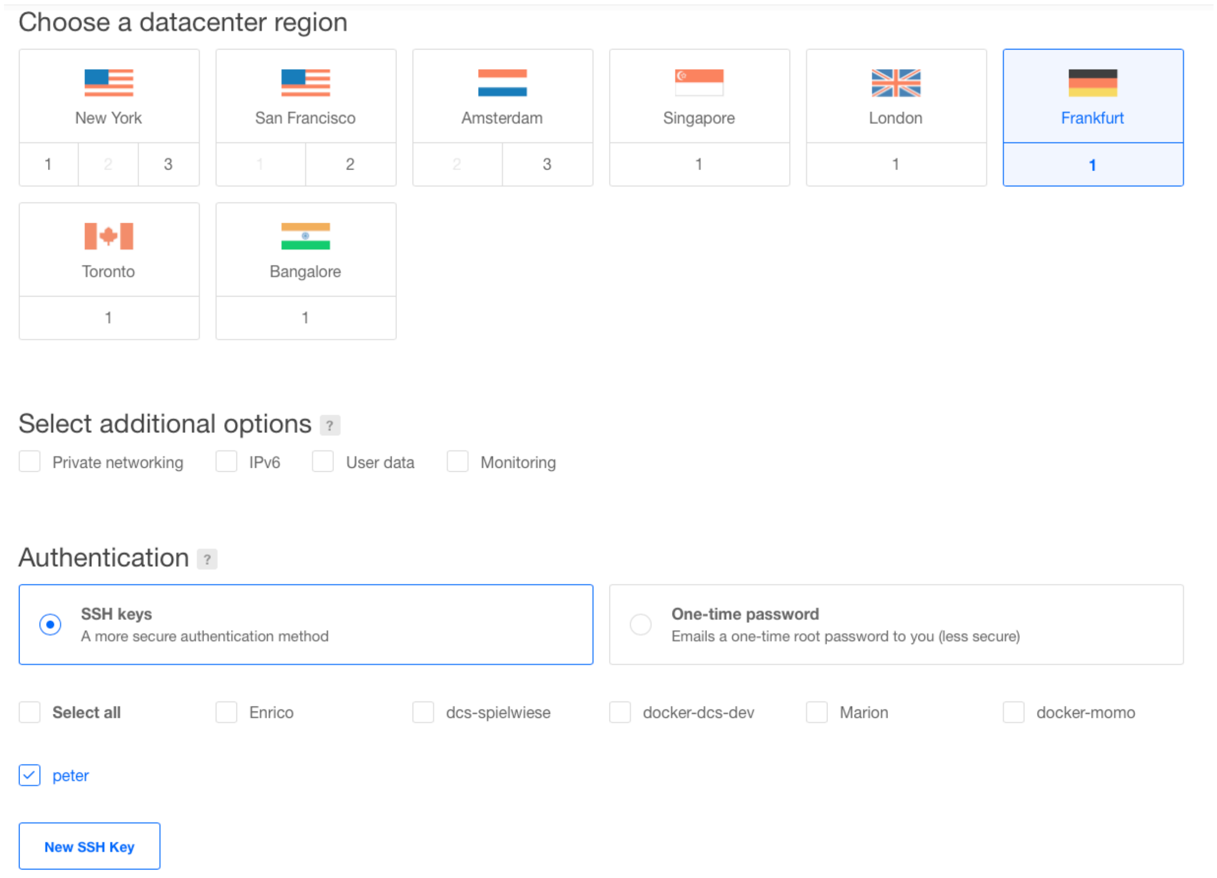
Here we choose a cheap (10$/month) server with ubuntu 18.04 / 2 GB RAM and 1 CPU



Standard virtual machines with a mix of memory and compute resources. Best for small projects that can handle variable levels of CPU performance, like blogs, web apps and dev/test environments.

$5/mo	$10/mo	$15/mo	$15/mo	$15/mo	$20/mo
$0.007/hour	$0.015/hour	$0.022/hour	$0.022/hour	$0.022/hour	$0.030/hour
1 GB / 1 CPU	2 GB / 1 CPU	3 GB / 1 CPU	2 GB / 2 CPUs	1 GB / 3 CPUs	4 GB / 2 CPUs
25 GB SSD disk	50 GB SSD disk	60 GB SSD disk	60 GB SSD disk	60 GB SSD disk	80 GB SSD disk
1000 GB transfer	2 TB transfer	3 TB transfer	3 TB transfer	3 TB transfer	4 TB transfer

Make sure the server is in Germany (or at least Europe) and that you can use your SSH key to login



Choose a datacenter region

New York	San Francisco	Amsterdam	Singapore	London	Frankfurt
1 2 3	1 2	2 3	1	1	[✓ 1]
Toronto	Bangalore				
1	1				

Select additional options ?

☐ Private networking ☐ IPv6 ☐ User data ☐ Monitoring

Authentication ?

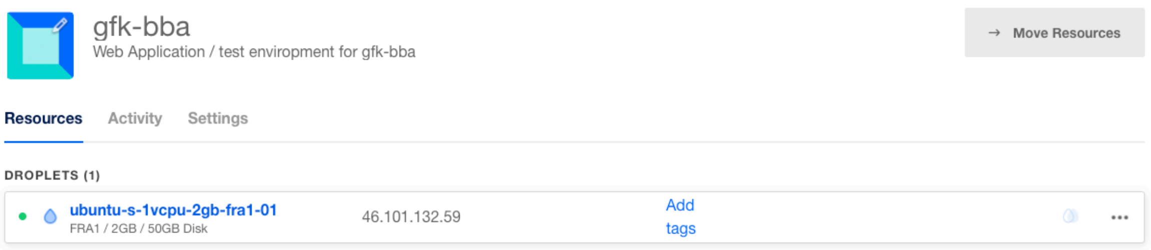
[✓ SSH keys
A more secure authentication method]

One-time password
Emails a one-time root password to you (less secure)

☐ Select all ☐ Enrico ☐ dcs-spielwiese ☐ docker-dcs-dev ☐ Marion ☐ docker-momo

☑ peter

After a minute or two we have the server up and running.
Copy the ip address (in my case 46.101.132.59)



gfk-bba
Web Application / test enviropment for gfk-bba

DROPLETS (1)

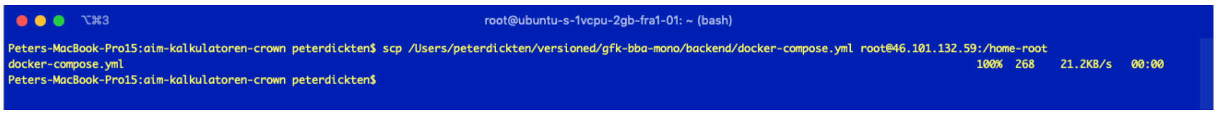
ubuntu-s-1vcpu-2gb-fra1-01 FRA1 / 2GB / 50GB Disk	46.101.132.59	Add tags

Now copy the docker-compose.yml file to the server

```
scp PATH_TO_FILE/docker-compose.yml root@SERVER_IP_ADDRESS:/root/
```

or in my example

```
scp /Users/peterdickten/versioned/gfk-bba-mono/backend/docker-compose.yml
root@46.101.132.59:/root/
```



```
root@ubuntu-s-1vcpu-2gb-fra1-01: ~ (bash)
Peters-MacBook-Pro15:aim-kalkulatoren-crown peterdickten$ scp /Users/peterdickten/versioned/gfk-bba-mono/backend/docker-compose.yml root@46.101.132.59:/home-root
docker-compose.yml                                                                                          100%  268    21.2KB/s   00:00
Peters-MacBook-Pro15:aim-kalkulatoren-crown peterdickten$
```

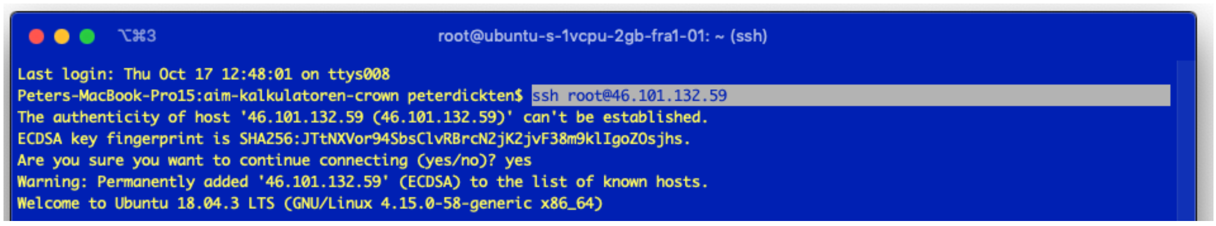
If the connect fails, make sure that you provided your public SSH key to your hosting provider

Login to the server:

```
ssh root@SERVER_IP_ADDRESS
```

or in my example

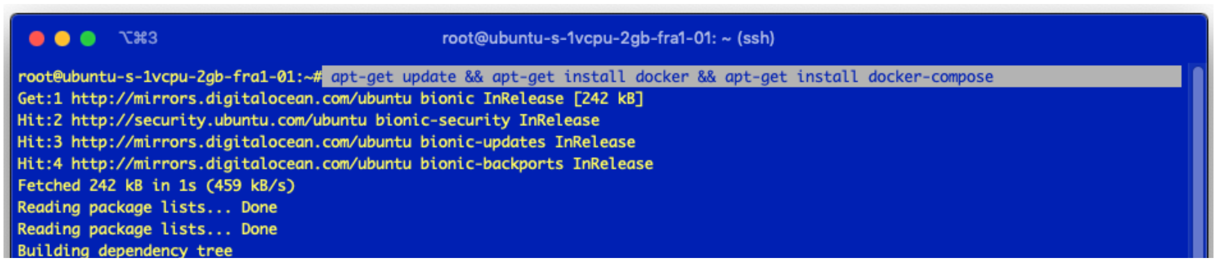
```
ssh root@46.101.132.59
```



```
root@ubuntu-s-1vcpu-2gb-fra1-01: ~ (ssh)
Last login: Thu Oct 17 12:48:01 on ttys008
Peters-MacBook-Pro15:aim-kalkulatoren-crown peterdickten$ ssh root@46.101.132.59
The authenticity of host '46.101.132.59 (46.101.132.59)' can't be established.
ECDSA key fingerprint is SHA256:JTtNXVor94SbsClvRBrcN2jK2jvF38m9klIgoZ0sjhs.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '46.101.132.59' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-58-generic x86_64)
```

Now we need docker + docker-compose on that server

```
apt-get update && apt-get install docker && apt-get install docker-compose
```



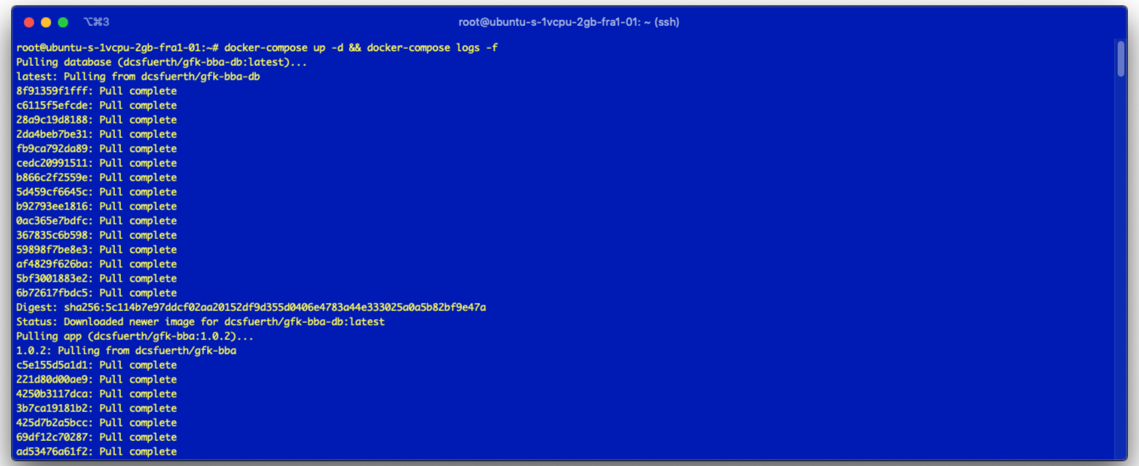
```
root@ubuntu-s-1vcpu-2gb-fra1-01: ~ (ssh)
root@ubuntu-s-1vcpu-2gb-fra1-01:~# apt-get update && apt-get install docker && apt-get install docker-compose
Get:1 http://mirrors.digitalocean.com/ubuntu bionic InRelease [242 kB]
Hit:2 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:3 http://mirrors.digitalocean.com/ubuntu bionic-updates InRelease
Hit:4 http://mirrors.digitalocean.com/ubuntu bionic-backports InRelease
Fetched 242 kB in 1s (459 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree
```

```
(and much more …)
```

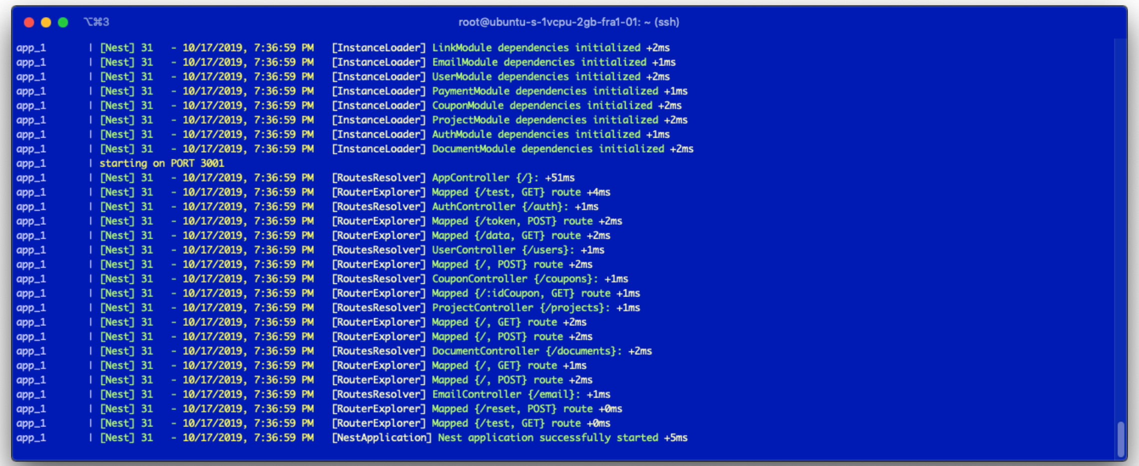
Now we have all what we need to run the application

```
docker-compose up -d && docker-compose logs -f
```

This will start the application in the background and show the logs.
You can stop the log output using [Ctrl] + [C]



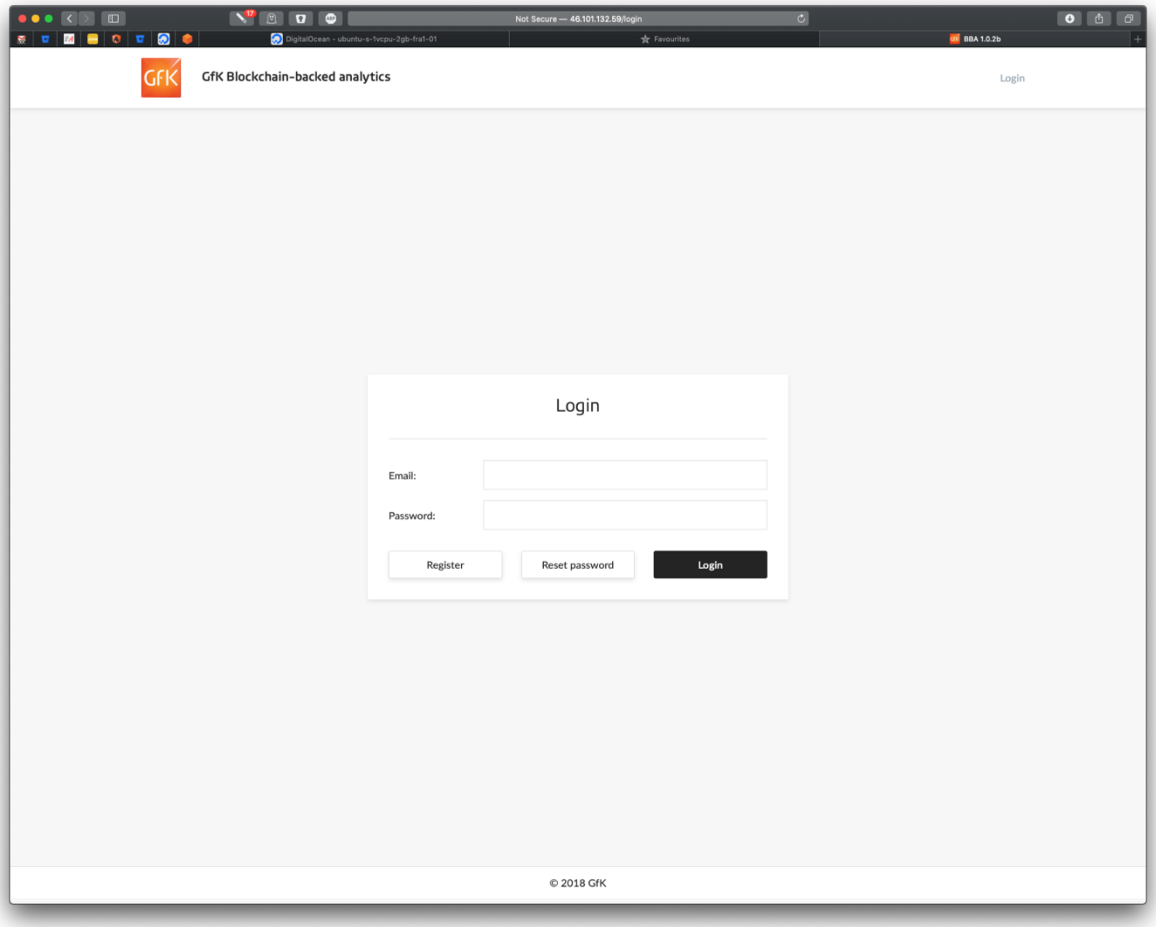
```
root@ubuntu-s-1vcpu-2gb-fra1-01:~# docker-compose up -d && docker-compose logs -f
Pulling database (dcsfuerth/gfk-bba-db:latest)...
latest: Pulling from dcsfuerth/gfk-bba-db
8f91359f1fff: Pull complete
c6115f5efcde: Pull complete
28a9c19d8188: Pull complete
2da4beb7be31: Pull complete
fb9ca792da89: Pull complete
cedc20991511: Pull complete
b866c2f2559e: Pull complete
5d459cf6645c: Pull complete
b92793ee1816: Pull complete
0ac365e7bdfc: Pull complete
367835c6b598: Pull complete
59898f7be8e3: Pull complete
af4829f626ba: Pull complete
5bf3001883e2: Pull complete
6b72617fbdc5: Pull complete
Digest: sha256:5c114b7e97ddcf02aa20152df9d355d0406e4783a44e333025a0a5b82bf9e47a
Status: Downloaded newer image for dcsfuerth/gfk-bba-db:latest
Pulling app (dcsfuerth/gfk-bba:1.0.2)...
1.0.2: Pulling from dcsfuerth/gfk-bba
c5e155d5a1d1: Pull complete
221d80d00ae9: Pull complete
4250b3117dca: Pull complete
3b7ca19181b2: Pull complete
425d7b2a5bcc: Pull complete
69df12c70287: Pull complete
ad53476a61f2: Pull complete
```



```
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [InstanceLoader] LinkModule dependencies initialized +2ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [InstanceLoader] EmailModule dependencies initialized +1ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [InstanceLoader] UserModule dependencies initialized +2ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [InstanceLoader] PaymentModule dependencies initialized +1ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [InstanceLoader] CouponModule dependencies initialized +2ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [InstanceLoader] ProjectModule dependencies initialized +2ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [InstanceLoader] AuthModule dependencies initialized +1ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [InstanceLoader] DocumentModule dependencies initialized +2ms
app_1        | starting on PORT 3001
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [RoutesResolver] AppController {/}: +51ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [RouterExplorer] Mapped {/test, GET} route +4ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [RoutesResolver] AuthController {/auth}: +1ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [RouterExplorer] Mapped {/token, POST} route +2ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [RouterExplorer] Mapped {/data, GET} route +2ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [RoutesResolver] UserController {/users}: +1ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [RouterExplorer] Mapped {/, POST} route +2ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [RoutesResolver] CouponController {/coupons}: +1ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [RouterExplorer] Mapped {/:idCoupon, GET} route +1ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [RoutesResolver] ProjectController {/projects}: +1ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [RouterExplorer] Mapped {/, GET} route +2ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [RouterExplorer] Mapped {/, POST} route +2ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [RoutesResolver] DocumentController {/documents}: +2ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [RouterExplorer] Mapped {/, GET} route +1ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [RouterExplorer] Mapped {/, POST} route +2ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [RoutesResolver] EmailController {/email}: +1ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [RouterExplorer] Mapped {/reset, POST} route +0ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [RouterExplorer] Mapped {/test, GET} route +0ms
app_1        | [Nest] 31   - 10/17/2019, 7:36:59 PM   [NestApplication] Nest application successfully started +5ms
```

The last line “Nest application successfully started” show that the app is running.

You can see the application running in the browser at http://SERVER_IP_ADDRESS or in my case http://46.101.132.59


Not Secure — 46.101.132.59/login
GfK Blockchain-backed analytics
Login
Login
Email:
Password:
Register
Reset password
Login
© 2018 GfK

Feel free to add a domain and/or SSL